International Conference on Instrumentation, Communication and Information Technology (ICICI) 2005 Proc.,
August 3-5, 2005, Bandung, Indonesia

Digital Control Signal Processor On FPGA Chip

Faizal A. Samman[1)], Isni Sumsulawati, Rahmi, Rhiza S. Sadjad[2)]
Department of Electrical Engineering, The University of Hasanuddin
Jl. Perintis Kemerdekaan Km.10 Makassar 90245. Phone/Fax: +62-411-588111.
e-mail: [1)]faizalas@unhas.ac.id, [2)]rhiza@unhas.ac.id

Abstract *– This paper presents the implementation of digital control signal processor (DCSP) on field programmable gate array (FPGA) chip. This research proposed novel design method to implement DCSP on FPGA chip. The DCSP structure follows digital PID controller block structure, where controller parameters: proportional, integral, and differential parameters can be easily reconfigured. Thus for any specific problem, the designers find no difficult to establish DCSP into VHDL-synthesis model and then into embedded digital electronic circuit. The DCSP has been successfully implemented and tested on Xilinx Spartan 2 XC2S50 FPGA development board. And it woks in 10 msec external ADC sampling period with 16-bit input (8-b decimal, 8-b decimal point) and 24-b output (8-bit decimal, 16-b decimal point). And maximum internal system clock is 198.216 MHz. DCSP uses 235 slices (logic cells) of 768 slices in XC2S50 chip. The estimated power consumption is 14 mW. For future works, this VHDL model of DCSP could be synthesized into physical circuit layout, and embedded on a single chip together with other blocks such as ADCs, DACs, PROM, SRAM, and other IP blocks. This work would be called Embedded DCSP System-on-Chip (SoC).*

Keywords *– Digital Control, Digital Signal Processing, PID Controller, FPGA, VHDL, Embedded Controller.*

I. INTRODUCTION

Implementation of embedded digital controller is the best choice to meet real-time solution, and system reliability. The problem in digital PID controller is how to cover binary-floating point information in the control signals. The binary-floating point values represented in number of bits in the inputs and outputs affects the controller performance. Digital electronic controller such as microcontroller has fixed number of I/O bits. Therefore, the designer should use more than one I/O ports to meet bit number of the controller I/Os. This work proposes a design methodology using Xilinx System-Generator™, and it provides good solution for this problem.

Several works have also used FPGA (field programmable gate array) device [1,4,5] or DSP (digital signal processor) device [3,4] to implement the digital controller. However, this work uses different approach, where the procedure starting from system modeling in block diagram. Figure 1 show feedback control system using DCSP block.

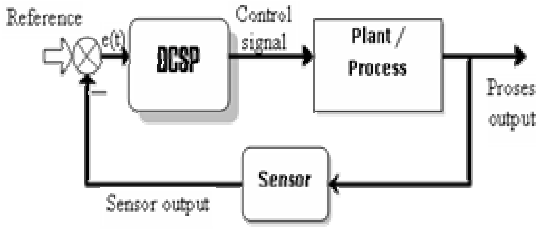


Fig.1. Control system with DCSP.

The design procedure to establish DCSP is:

1. Continuous-time model of PID is designed to any specific plant model. Then it is transferred into discrete-time PID.
2. The system is verified by simulation to observe the performance. If performance is satisfied, the block is transferred into SystemGenerator™ discrete-time block. This block is then called DCSP.
3. The VHDL model of DCSP is established using SystemGenerator™. Then it is compiled to obtain configuration file.
4. Several floating point test inputs are generated to verify DCSP block outputs in Matlab, then converted them into binary-floating point as test vectors to verify the responses of the VHDL model of DCSP.

5. If the results of point 4 are stisfied, the configuration file obtained in point 3 is downloaded into Spartan 2 XC2S50 FPGA chip.
6. The core is then verified by testing.

Figure 2 present the design procedure of DCSP starting from SystemGenerator block design until circuit routing of the DCSP in Spartan 2 FPGA chip.

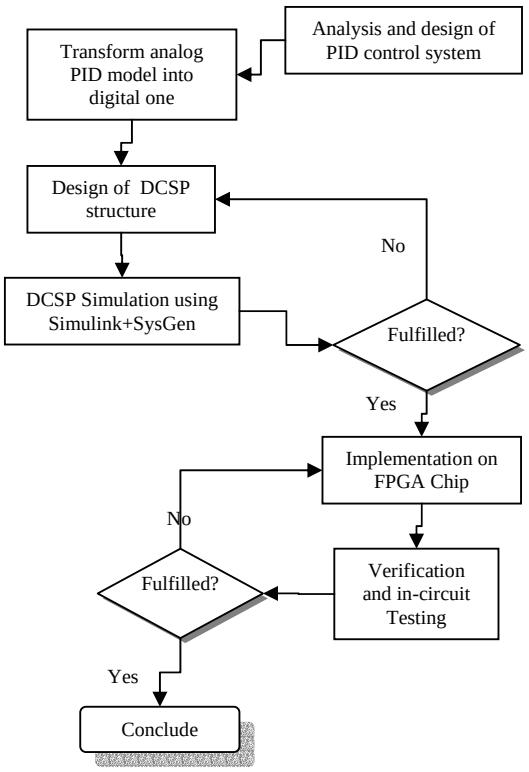


Fig.2. Design method of DCSP using Xilinx SystemGenerator.

II. DIGITAL CONTROLLER DESIGN

The discrete-time mathematical model of DCSP is derived from its continuous-time model as follow.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \qquad (1)$$

In Laplace Transform,

$$U(s) = K_p E(s) + \frac{K_i}{s} E(s) + K_d s E(s) \qquad (2)$$

And in transfer function model is

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \qquad (3)$$

Some reference [6,7] derive PID controller structure (*discrete-time model*) in the following (T is sampling period).

$$u(k) = K_p e(k) + K_i T\, S(k) + K_d \frac{e(k) - e(k-1)}{T} \qquad (4)$$
$$S(k) = S(k-1) + \frac{T}{2}(e(k) + e(k-1))$$

Variable S(k) above is a sum variable that describes integration process (summation) of current and in advance signals. Transfer Function in Z-Transform of equations (4) is

$$D(z) = \frac{U(z)}{E(z)} = K_p + \frac{K_i T}{2}\left(\frac{z+1}{z-1}\right) + \frac{K_d}{T}\left(\frac{z-1}{z}\right) \qquad (5)$$

Some references also present discrete-time PID model in the following.

$$u(k) = K_p e(k) + K_i T \sum_{n=0}^{k} e(n) + \frac{K_d}{T}(e(k) - e(k-1)) \qquad (6)$$

Equation above is for sampling at time *k*, thus for sampling at *k-1*,

$$u(k-1) = K_p e(k-1) + K_i T \sum_{n=0}^{k-1} e(n) + \frac{K_d}{T}(e(k-1) - e(k-2)) \qquad (7)$$

Substacting (2.5) from (2.6) yields

$$u(k) = u(k-1) + K_p(e(k) - e(k-1)) + K_i T e(k) + \frac{K_d}{T}(e(k) - 2e(k-1) + e(k-2)) \qquad (8)$$

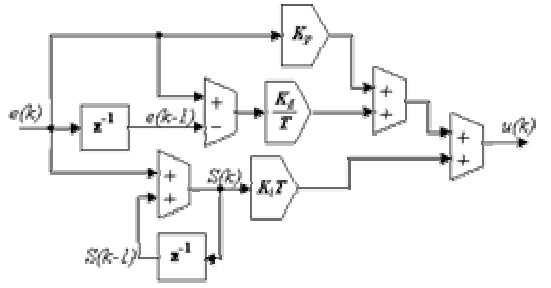
Transfer function of (8) is

$$D(z) = \frac{U(z)}{E(z)} = K_p + K_i \frac{Tz}{z-1} + K_d \frac{z^2 - 2z + 1}{Tz(z-1)} \qquad (9)$$

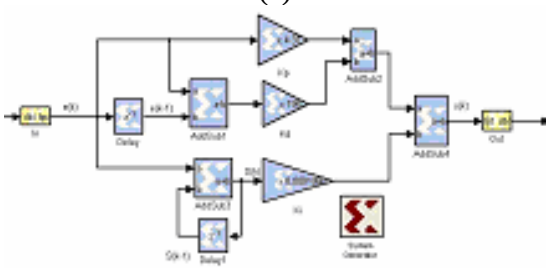
By factorizing $(z^2 - 2z + 1)$ by $(z - 1)^2$ then yielding

$$D(z) = \frac{U(z)}{E(z)} = K_p + K_i \frac{Tz}{z-1} + K_d \frac{z-1}{Tz} \qquad (10)$$

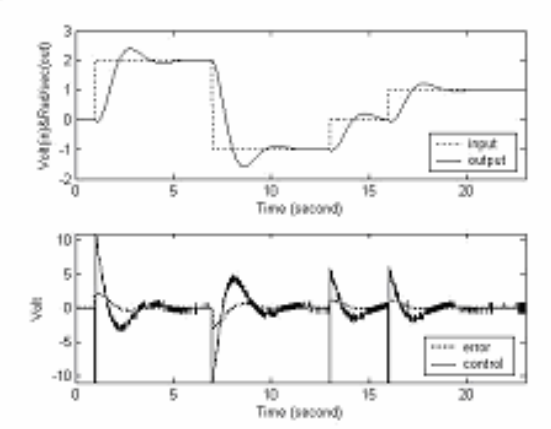
Figure 3(a) shows the digital PID controller structure described from mathematical model in equation (10).



(a)



(b)



(c)

Fig.3. (a) Discrete-time PID controller structure, and its (b) model using SystemGenerator blocks. (c) System response (system input, output, error and control signal.

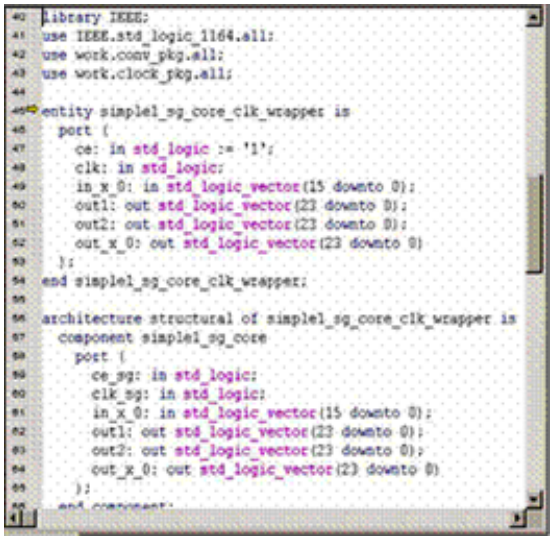
After the digital PID controller is designed using Simulink, then it simulated to see its performance. If the performance is satisfied, the block in Simulink is redesigned using SystemGenerator blocks as shown in Figure 3(b). In this design example, it uses K_p=4.37, K_i=0.01 and K_d=1.95. And with sampling period T=0.01 sec, then the discrete-time K_i and K_d change to K_i=0.01*0.01=0.0001, and K_d=1.95/0.01=0.195. While the K_p value is not change.

Using all parameters the system is simulated to see its performance to control double-integrator plant. Figure 3(c) shows the simulation results. Upper diagram show the responses of the reference input (dashed-line) and the system output (solid-line). Lower diagram shows system error signal or input signal into DCSP (dashed-line), and control signal produced by DCSP (solid-line). It looks that the control signal is steady while the error goes to zero value (no system error).

Once again, the system is simulated to compare the performance resulted using Simulink blocks. If the performance is close similar, then VHDL code is generated. The diagram in Figure 3(b) is then called DCSP (digital control signal processor) block diagram.

III. DCSP IMPLEMENTATION ON FPGA CHIP

After building DCSP block diagram using SystemGenerator blocks, the VHDL code of DCSP is generated. Figure 4 exhibits the top list of the VHDL code.

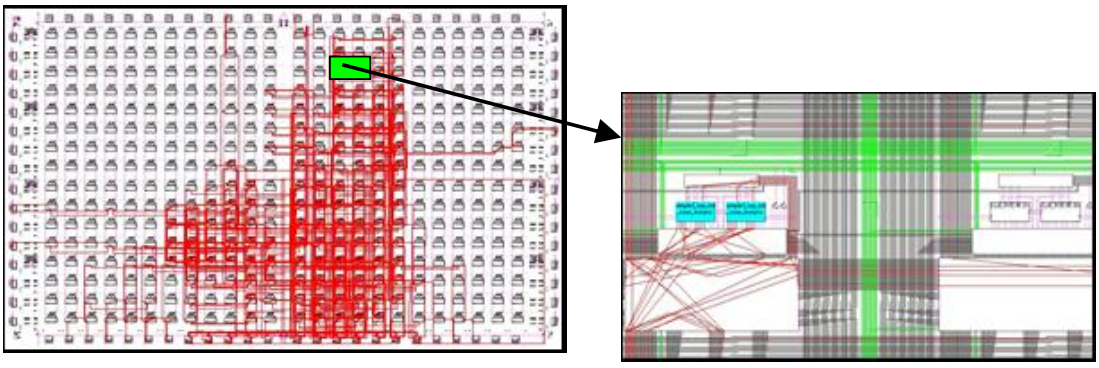


```
library IEEE;
use IEEE.std_logic_1164.all;
use work.conv_pkg.all;
use work.clock_pkg.all;

entity simple1_sg_core_clk_wrapper is
  port (
    ce: in std_logic := '1';
    clk: in std_logic;
    in_x_0: in std_logic_vector(15 downto 0);
    out1: out std_logic_vector(23 downto 0);
    out2: out std_logic_vector(23 downto 0);
    out_x_0: out std_logic_vector(23 downto 0)
  );
end simple1_sg_core_clk_wrapper;

architecture structural of simple1_sg_core_clk_wrapper is
  component simple1_sg_core
    port (
      ce_sg: in std_logic;
      clk_sg: in std_logic;
      in_x_0: in std_logic_vector(15 downto 0);
      out1: out std_logic_vector(23 downto 0);
      out2: out std_logic_vector(23 downto 0);
      out_x_0: out std_logic_vector(23 downto 0)
    );
  end component;
```

Fig.4. VHDL-code of DCSP (entity declaration on top list).

The VHDL code is compiled to see the implementation result on XC2S50 chip. Figure 5(a) shows the circuit routing of DCSP on XCS50 FPGA chip. Figure 5(b) shows the detailed view of the CLB in row 3

and column 15 and 16 on the chip. It looks that CLB in row 3 column 16 is not used. One CLB consists of two slices. A slice is a logic cells containing LUT (look-up table) components, Flipflops, and any other components on XC2S50.



(a) (b)

Fig.5. (a) Circuit routing of DCSP on Xilinx Spartan 2 XC2S50 FPGA chip. (b) Detail view of the CLB (configurable logic block) in row 3, column 15 and 16.

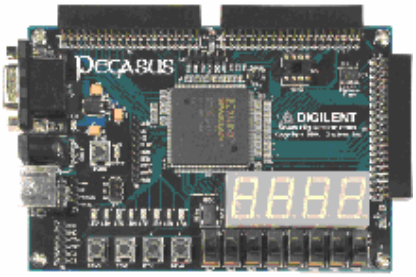


Fig.6. Pegasus Digilent Board with Xilinx Spartan 2 XC2S50.

Table 1 concludes the logic gates used to implement DCSP. It uses 235 of 768 slices, or it consumes about 30 % of total slices in the XC2S50 device. The DCSP uses internal system clock, and it is suggested to use minimum clock period 5.045ns or maximum clock frequency 198.216MHz.

Component :	Used	Out of	Percent
Slices	155	768	20 %
Slice Flipflops	16	1536	1 %
4-input LUTs	301	1536	19 %
Bonded IOBs	40	140	28 %
GCLK	1	4	25 %

Table 1. Implementation result using XCS50pq208-6 device.

IV. SIMULATION AND TESTING RESULTS

The DCSP is simulated using some test-bench. Figure 7, 8 and 9 show three examples of the testbench. It looks that the output is represented in hexadecimal and bit terms.

To verify the DCSP functional, three functional simulations are performed. The first is simulation result of the DCSP using Simulink block.

The second one is simulation result of one using SystemGenerator block. The test pattern used in the first simulation is again used in this simulation.

And the last is simulation result using testbench using Xilinx ISE Foundation software as shown in Figure 7, 8 and 9. About 120 test-vectors have been used to test the DCSP using Spartan 2 XC2S50 development board (donated by Digilent Inc.) as shown in Figure 6. The test vectors are translated from test patterns used in advanced simulation (decimal point data) into binary test vectors.

The configuration file is then downloaded into XC2S50 and tested on Pegasus Board (see Figure 6) using seven-segments, LEDs, switches, and pushbutton attached on the board.

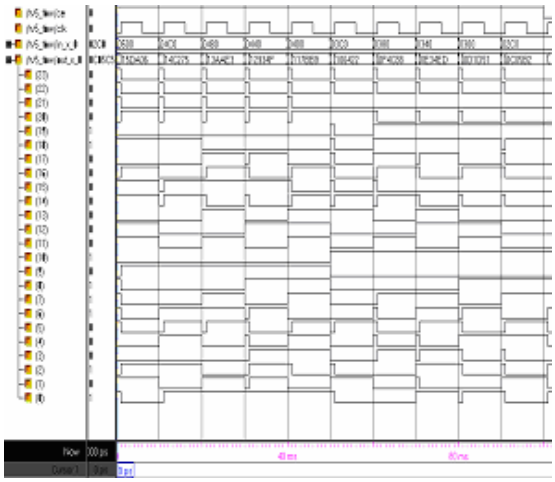


Fig.7. Timing diagram of DCSP I/O simulation. with test-vector 1.

Fig.8. Timing diagram of DCSP I/O simulation. with test-vector 2.

Fig.9. Timing diagram of DCSP I/O simulation with test-vector 3.

V. CONCLUSION

. The DCSP has been successfully implemented and tested on Xilinx Spartan 2 XC2S50 FPGA development board. And according to the discrete-time model of the DCSP, it woks in 10 msec external ADC sampling period. The implementation result of the DCSP is concluded in the following:

1. DCSP has 16-bit input (8-b decimal, 8-b decimal point) and 24-b output (8-bit decimal, 16-b decimal point).
2. Maximum internal system clock is 198.216 MHz or minimum internal sampling period is 5.045ns.
3. DCSP consumes 235 slices (logic cells) of 768, or 20 % of total slices in XC2S50 chip.
4. The total estimated power consumption is about 14 mW.
5. About 120 test-vectors are used to simulate the timing and functional behavior of the DCSP. And it has provide similar result to functional simulation using Simulink.
6. The DCSP has been tested using Spartan 2 development board. And the testing has given similar results to functional simulation using Xilinx ISE Foundation software development. Thus the design has fulfilled required functional and performance.

For future works, this VHDL model of DCSP could be synthesized into physical circuit layout, and embedded on a single chip together with other blocks such as ADCs, DACs, PROM, SRAM, and other IP blocks. This work would be called Embedded DCSP System-on-Chip (SoC).

VI.ACKNOWLEDGEMENTS

Special thanks to Xilinx Corporation for University Donation of ISE Foundation development software, and Digilent Incorporation for Pegasus Spartan 2 XC2S50 development board.

VII.REFERENCES

[1] Joaquin Garcia, Gilberto Viveros, Rene Cumplido (2004). "FPGA-Based Architecture for Radar's STC, FTC and Gain Modules. *International Conference on Reconfigurable Computing and FPGAs*, Colima, Mexico, Sept.20-21, 2004.

[2] David Gwaltney, Ken King, Keary Smith (2003). "Implementation of Adaptive Digital Controllers on Programmable Logic Devices". Research Project, NASA Marshall Space Flight Center, Huntsville, AL, USA.

[3] H. Dong, R. Flood, C. Hovater, J. Musson (2002). "A Dual Digital Signal Processor VME Board For Instrumentation and Control Applications". Thomas Jefferson National Accelerator Facility, Newport News, VA, USA. Technical Report: DOE Contract No. DE-AC05-84-ER40150.

[4] Paul Graham, Brent Nelson (2000). "Reconfigurable Processors for High-Performance, Embedded Digital Signal Processing". Technical Report: Defense Advanced Research Projects Agency (DARPA) and Rome Laboratory, Air Force Material Command, USAF, under Agreement No.F30602-97-1-0222.

[5] C. Reiser, L. Znamirowski, O.A. Palusinski, S.B.K. Vrudhula, D. Rakhmatov (1998). "Dynamically Reconfigurable Analog/Digital Hardware Implementation Using FPGA and FPAA Technologies". Journal on Circuits, Systems and Computers, World Scientific Pub., Oct.1998.

[6] Charles L. Phillips, H. Troy Nagle (1995). Digital *Control System Analysis and Design*, 3rd ed. Prentice-Hall, New Jersey, 1995.

[7] Charles L. Philips, Royce D. Harbor (1996). *Feedback Control Systems*, 3rd ed. Prentice-Hall, New Jersey, 1996.

[8] Richard C. Dorf, Robert H. Bishop (2001). *Modern Control Systems*, 9th ed. Prentice-Hall, New Jersey 2001.

[9] Benjamin C. Kuo (1995). *Automatic Control Systems*, 7th ed. Prentice-Hall, New Jersey, 1995.